(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 555 816 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
20.07.2005 Bulletin 2005/29

(51) Int Cl.[7]: **H04N 5/46**

(21) Application number: **04000556.3**

(22) Date of filing: **13.01.2004**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR
Designated Extension States:
AL LT LV MK

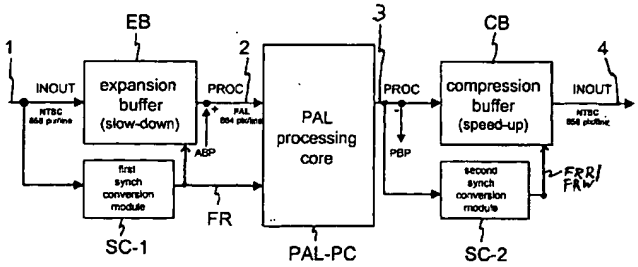
(71) Applicant: **Sony International (Europe) GmbH**
10785 Berlin (DE)

(72) Inventors:
- **Schwendowius, Jörg**
  c/o Stuttgart Technology Center
  70327 Stuttgart (DE)
- **Scholz, Alexander**
  c/o Stuttgart Technology Center
  70327 Stuttgart (DE)
- **Golov, Gil c/o Stuttgart Technology Center**
  70327 Stuttgart (DE)
- **Dilly, Altfried c/o Stuttgart Technology Center**
  70327 Stuttgart (DE)

(74) Representative: **Müller - Hoffmann & Partner Patentanwälte,**
Innere Wiener Strasse 17
81667 München (DE)

(54) Method for video signal processing

(57) The design of a video processing core designed for processing video signals of a specific video format requires a high expense. It is therefore desirable to be able to use such a video processing core for other video formats as well. The invention therefore provides the method for video signal processing, wherein the following steps are performed: receiving an input video signal (1) having a first or input format (INOUT, NTSC; INOUT, PAL), processing said input video signal (1) or a modified video signal (2), which is a derivative of said input video signal (1), using a video signal processing method, which is designed for processing video signals of a fixed second or processing format (PAL; NTSC) thus generating a processed video signal (3), outputting said processed video signal (3) or a derivative thereof as an output video signal (4), wherein said modified video signal (2) is derived from said input video signal (1) by adding or removing pixels to/from said input video signal (1), thereby obtaining said modified video signal (2) with an intermediate format (PROC), if said first or input format (INOUT, NTSC; INOUT, PAL) is not conform to said fixed or processing second format, wherein said output video signal (4) is derived from said processed video signal (3) by accordingly removing or adding pixels from/to said processed video signal (3) thereby obtaining an output video signal (4) of said first format, if said first or input format (INOUT, NTSC; INOUT, PAL) is not conform to said second or processing format (PAL; NTSC).



CIF

Fig. 1

Printed by Jouve, 75001 PARIS (FR)

EP 1 555 816 A1

Description

[0001] The invention relates to a method for video signal processing.

[0002] It is a complex task to implement video signal processing algorithms in hardware. Such hardware is typically referred to as video processing core. During the design process of a video processing core evaluation and verification are very important steps. Developing a video processing core, i. e. a core processing module for a single video format, e. g. PAL format or NTSC format, represents the lowest requirement and thus a fast and controllable design effort. Maintaining this single format design environment is desirable. It has to be noted that the format names "PAL" and "NTSC" throughout this description only denote the number of pixels per video line, the number of video lines per field and the field rate common to these formats as specified in the digital video format ITU-R 656. The names originate from the analogue format specifications and specify more parameters, such as colour coding, modulation schemes, and so on. But throughout this description only the mentioned parameters pixels per line, lines per field and fields per second will be referenced.

[0003] In other words, it is desirable to be able to use a video processing core designed for a specific video format for other video formats as well.

[0004] For this purpose, in prior art, a video signal of an input format that is different from the processing format of the processing core, i. e. the format the processing core is designed for, is completely converted into a video signal having said processing format. However, such a complete conversion has several disadvantages, e. g. the longer time needed for format conversion, additional needed hardware, as well as a differing temporal base for fields which influences motion rendering. Also, large memories and logic functions are needed to perform a full format conversion process.

[0005] It is an object underlying the invention to provide a method for video signal processing which enables the use of a video processing method designed for a specific processing format for processing a video signal having an input format which can be different from the processing format with very little expense.

[0006] To achieve this objective, the invention provides a method for video signal processing according to claim 1. In addition, the invention provides a signal processing system, a computer program product, and a computer readable storage medium as defined in claims 28, 32 and 33, respectively. Further features and preferred embodiments are respectively defined in respective sub claims and/or in the following description.

[0007] Within the inventive method for video signal processing, the following steps are performed: receiving an input video signal having a first or input format, processing said input video signal or a modified video signal, which is a derivative of said input video signal, using a video signal processing method, which is designed for processing video signals of a fixed second or processing format thus generating a processed video signal, outputting said processed video signal or a derivative thereof as an output video signal, wherein said modified video signal is derived from said input video signal by adding or removing pixels to/from said input video signal, thereby obtaining said modified video signal with an intermediate format, if said first or input format is not conform to said fixed second or processing format, wherein said output video signal is derived from said processed video signal by accordingly removing or adding pixels from/to said processed video signal thereby obtaining an output video signal of said first format, if said first or input format is not conform to said second or processing format.

[0008] Thereby, said video signal processing method is designed to work on a specific line width determined by the processing format, i. e. the core format. According to the invention the INOUT format line length, i. e. the format line length of the input video signal is changed in order to match the processing format line length, i. e. the format line length of the processing format. Further, the number of lines per video field to be processed by the video signal processing method, which may be performed by the core, i. e. the video processing core, is determined by a respective field restart signal, i. e. video synch signal, and not by the core format line number, i. e. the number of lines of the core format, i. e. the processing format.

[0009] For example, a PAL processing core can be interrupted and restarted before reaching 312 lines, whereas an NTSC processing core can process more than 262 lines. It should be noted, that the pixel rate of the INOUT format is identical to the pixel rate of the PROC format. Note, that the invention is also applicable with differing pixel rates. In this case only the sizes of the buffers need to be increased (for the use of buffers see below).

[0010] An aspect of the invention is therefore to add or remove pixels to/from said input video signal, such that the resulting video signal, i. e. said modified video signal, has the processing line length, i. e. the number of pixels per line of the processing format of the video processing core.

[0011] According to the invention said intermediate format may not be a valid standard video format, in particular it may neither be of the first or input format nor of the fixed second or processing format. This means, no format conversion process in the classical sense of the term is performed. Those skilled in the art will notice the difference. The modified video signal is therefore not conform with any valid standard video format, e. g. PAL standard or NTSC standard. It should be clearly understood at this point, that the intention of the invention is not to do format conversion.

[0012] In a first embodiment (cf. below) it is the target of the invention to add NTSC signal processing capability to

8/25/06, EAST Version: 2.1.0.14

a signal processing core unit, i. e. above mentioned video processing core, which was originally designed for processing PAL video signals. This, however, does not mean that internally the NTSC signal is converted to a valid PAL format. Instead, the NTSC signal is internally handled the same as the PAL signal. Therefore, additional video information is stored in extra memory space (cf. below). As the NTSC format compared to the PAL format contains more video fields and less video lines per field, the video processing in the core module is prematurely interrupted and restarted more often than during PAL processing. This means, there are different control state machines needed for PAL and NTSC processing, as the field restart signals of the processing core module are different for PAL and NTSC. The internal core processing signal used in the NTSC mode is just similar to PAL, but not a valid PAL format. As already mentioned, a processing core which was originally designed for PAL video signals only is used. According to the invention, the processing capability is extended to NTSC signal by adding some additional memory space and additional control logic, cf. first embodiment below. Again, format conversion from PAL to NTSC standard and vice versa is not possible with the invention. A PAL input produces a PAL output according to the first embodiment below, and an NTSC input produces an NTSC output as is the case for a second embodiment below. A benefit of the invention is the easy adaptation of an existing and tested PAL system processing core to NTSC format (cf. first embodiment below) or vice versa (cf. second embodiment below).

[0013] According to the invention, the number of lines of said intermediate format is equal to the number of pixels per video field of said first or input format divided by the number of pixels per video line of said second or processing format. As mentioned, the invention is based on the assumption of an equal number of processed pixels per time instance, i. e. an equal pixel rate for both formats, i. e. the input format and the processing format. But as mentioned above, a differing input and output pixel rate compared to the core processing pixel rate is possible but not further considered throughout the remainder of this description.

[0014] In said first embodiment the source format NTSC has 858 pixels per line and 263 lines per field for a long field, which equals 225654 pixels per field. The core processing format in this first embodiment is the PAL format and has 864 pixels per line. This means, the number of lines of said intermediate format is given by 225654 / 864 = 261.1 lines. This includes all required 244 lines of active video in NTSC format.

[0015] In the second embodiment the input video signal is a PAL format signal and the processing core format is the NTSC format. In this case 864 pixels per line and 313 lines per field, i. e. per long field, which equals 270432 pixels per field need to be processed by the NTSC core. In this case, the number of lines of said intermediate format equals 270432 / 858 = 315.1 lines. These are actually more lines than the NTSC format specifies. Thus, the processing core must not be limited in the number of lines to be processed between successive input fields. It should be noted at this point, that the temporal field rate and the number of lines per field are interchangeable parameters in the respective format. With an equal field rate for both formats, wider lines mean fewer lines per field or vice versa. Also, with an equal number of lines per field for both formats, wider lines mean a lower field rate, respectively.

[0016] According to the invention, said first or input format is preferably a first standard video format, e. g. NTSC-/PAL-format, and/or said fixed second or processing format is a second standard video format, e. g. PAL-/NTSC-format. In the first embodiment of the invention, said first standard video format is the NTSC format and said second standard video format is the PAL format.

[0017] Preferably, when deriving said modified video signal from said input video signal pixels are added/removed so as to change the number of pixels per video line of said intermediate format with respect to said first or input format. Thereby, no other modifications of the input video signal are performed, only pixels are added or removed. Therefore only very little expense is needed.

[0018] Preferably, pixels are added/removed to/from said input video signal such that the number of pixels per video line of said intermediate format is equal to the number of pixels per video line of said fixed second or processing format. As mentioned above, this is because a processing core is designed to work on a specific line width which is determined by the core format i. e. the processing format. In the first embodiment six pixels per line are added so as to change the number of pixels per line of the input video signal which is an NTSC signal and therefore has 858 pixels per line to 864 pixels per line, which is conform with the processing format, which is the PAL format.

[0019] Preferably, when deriving said output video signal from said processed video signal said pixels are removed/added so as to change the number of pixels per video line of said output video signal with respect to said processed video signal.

[0020] In particular, pixels are removed/added from/to said processed video signal such that the number of pixels per video line of said output video signal is equal to the number of pixels per video line of said first or input format. For example, in the first embodiment 6 pixels per line are removed from the processed video signal, thus generating an output video signal of the NTSC format.

[0021] Further, said video signal processing method preferably uses the field restart signal of said input video signal, such that the field processing rate of said processing method is equal to the field rate of the first or input format. This means, processing is done field-wise and the processing of a video field is governed by the field restart signal of the input video signal. It should be noted in this context, that this means that the field processing rate of the video processing

8/25/06, EAST Version: 2.1.0.14

core is determined by the field rate of the input video signal. A field restart, i. e. a field restart signal is a certain trigger condition derived from the embedded synch signals or internally generated synch signals. E. g., the beginning of the first active video line in a field can be a field restart pulse. The actual restart condition is arbitrary and depends on the implementation. It only has to be the same condition for all fields, whether coming in from external or being generated internally as a result of a video data processing step. In general, a field restart signal indicates a start condition, which occurs only once per video field, preferably at the beginning of a field.

[0022] As already mentioned, said video processing is preferably performed by a video processing core, which is in particular adapted/designed for said fixed second or processing format, i. e. the video processing core is adapted/designed for processing video signals of said fixed second or processing format. The invention enables the use of this video processing core together with an input video signal of a different format than the processing format, i. e. the first or input format. Thereby, no changes are necessary within the video processing core.

First Embodiment

[0023] In the first embodiment said first or input format is the NTSC standard video format and said second or processing format is the PAL standard video format.

[0024] In this first embodiment preferably blank pixels are added at the end of an input video signal line, thus obtaining said modified video signal having said intermediate format. Blank pixels may be seen as pixels that do not contain any information.

[0025] Also, in this first embodiment pixels of said input video signal are preferably stored in a first buffer and said processing method reads pixels from said first buffer. This first buffer can be referred to as an expansion buffer or slow-down buffer, as it is used to extend the line width of the input video signal to match the line width of the second or processing format. Pixels of the input video signal are stored, i. e. written to said first buffer at the same time the video processing core reads pixels from the first buffer. After one video line of said input video signal has been written to the buffer and due to the same pixel rate of the video processing core has been processed, pixels are added, i. e. 6 pixels, which will be processed by the video processing core subsequently. While processing these added pixels further pixels of the input video signal arrive and are stored in said first buffer. The buffer may be seen as a First-In-First-Out (FIFO) buffer. Also, the input video signal may be regarded as an input video stream consisting of pixels that are written to the first buffer. At the output of the first buffer, pixels are added into the input video stream after the end of a video line of the input video stream. These pixels are also processed by the processing core and subsequently arriving pixels of the input video stream are stored in the expansion buffer during the processing of the added pixels by the video processing core. This way said first buffer will continually fill up during the processing of one video field, i. e. because of the unequal write/read ratio to/from said first buffer.

[0026] Preferably, processed pixels of said processed video signal which correspond to added pixels of said modified video signal are removed from said processed video signal, thus obtaining said output video signal. In other words, the blank pixels that have been added at the end of each input video signal line and that have been processed by the processing method are removed. This means, no special care must be taken by the video processing core when processing the blank pixels as the processing pixels are removed after the processing. The blank pixels should be of values such, that the processing core will not be disturbed or interrupted. Repetition of the last valid source pixel value could be a preferred method to derive a suitable blank pixel value.

[0027] Advantageously, pixels of said processed video signal are stored in a second buffer and said output video signal is generated by reading pixels from said second buffer. This second buffer may be referred to as compression buffer or speed-up buffer, as it is used for removing processed blank pixels. The second buffer may also be regarded as a FIFO-buffer, wherein the video processing core writes processed pixels to said second buffer which are then output on a FIFO-basis, thus generating said output video signal. Only IO-pixels will be written. The blank pixels will be omitted from writing into the FIFO-buffer, as they will be discarded anyways.

[0028] Preferably, a first buffer and/or a second buffer are used having a capacity to at least store a minimum number of pixels, which is determined by the number of added pixels per video field of said input video signal. The needed maximum buffer capacity will be computed later. It depends also on whether only active video is stored or an entire video line (see below).

[0029] Reading of pixels from said second buffer does not start until said second buffer is filled with at least said minimum number of pixels. The functioning of the second buffer can be better understood with the following elucidations: In the first embodiment on the output side of the processing core, the PAL signal has to be reformatted to match the first or input format, i. e. the NTSC format. Writing into the compression buffer starts as soon as the processing module provides valid data, i. e. processed video pixels. When the compression buffer is filled with said minimum number of pixels, reading starts with a higher rate. It can be advantageous to dimension the buffer such that the maximum number of pixels that can be stored in the buffer is equal to said minimum number of pixels. In this case reading starts with the higher rate when the compression buffer is completely filled up. In the first embodiment for every 864 pixels written,

8/25/06, EAST Version: 2.1.0.14

858 pixels are read, and 6 pixels are discarded. When the entire buffer is empty, the NTSC field has been written out. This concludes one full cycle, i. e. one processed video field.

Second Embodiment

[0030] In the second embodiment of the invention said first or input format is the PAL standard video format and said second or processing format is the NTSC standard video format.

[0031] In this second embodiment pixels are preferably removed at the end of an input video signal thus obtaining said modified video signal having said intermediate format. The pixels that are removed are pixels corresponding to Hblank and/or Vblank data. In other words, no pixels corresponding to active video data are removed.

[0032] Preferably, pixels to be removed correspond to pixels of the horizontal blanking period. However, this depends on the possibly embedded auxiliary data contained in the Hblank data. A person skilled in the art will know which data can be safely skipped, i. e. removed.

[0033] Preferably, pixels of said input video signal are stored in a second buffer and said processing method reads pixels from said second buffer. This second buffer may be referred to as a compression buffer or speed-up buffer as pixels are skipped from the input video signal. It should be noted that reading from the compression buffer does not start until sufficient data, which correspond to the number of pixels that will be skipped during the processing of one video field, are in the compression buffer. It has to be noted that pixels to be skipped will not be written to the buffer.

[0034] Further, in this second embodiment blank pixels are added at the end of each video line of said processed video signal, thus obtaining said output video signal of said first or input format. This method is equal to the input stage of the first embodiment. Added blank pixels must not disturb the outgoing video stream, e. g. by adding synch patterns.

[0035] Also, in this second embodiment pixels of said processed video signal are stored in a first buffer and said output video signal is generated by reading pixels from said first buffer. This first buffer may be referred to as expansion buffer or slow-down buffer because it is used to store pixels that arrive from the video processing core during the time blank pixels are added at the output of said first buffer.

[0036] Advantageously, a second buffer and/or a first buffer are used having a capacity to at least store a minimum number of pixels, which is equal to the number of pixels removed from said input video signal per video field.

[0037] Preferably, said processing method does not start reading pixels from said second buffer until said second buffer is filled with at least a minimum number of pixels. This way, at the end of the processing of one video, the compression buffer will be empty.

[0038] Also, said processing method may not start reading pixels from said second buffer until a field restart read signal is provided.

[0039] A signal processing system according to the invention is capable of performing or realizing a method for video signal processing as defined above and/or the steps thereof.

[0040] The inventive signal processing system preferably comprises a receiving means adapted for receiving an input video signal, an adding/ subtracting means adapted for adding/subtracting pixels to/from said input video signal thus generating a modified video signal having an intermediate format, a processing means designed for processing video signals of a fixed second or processing format, which processing means is adapted for processing said modified video signal, and thus generating a processed video signal, a subtracting/adding means adapted for subtracting/adding pixels from/to said processed video signal thus generating an output video signal of said first format, and an outputting means adapted for outputting said output video signal. It should be noted that "designed for" here means that the purpose or destiny of the processing means, e. g. a hardware video processing core, is to process video signals of said fixed second or processing format. Without the invention, it is not possible to use said processing means for processing video signals of a format that is different from said fixed second or processing format. As mentioned, the design of a processing means in hardware requires a high expense. With the invention, a once designed hardware processing core for a fixed second or processing format can be used without any modifications for processing video signals of a different format than said fixed second or processing format.

[0041] The inventive signal processing system further preferably comprises at least one storage means adapted for storing at least parts of said input video signal and/or at least parts of said processed video signal. Writing to and reading from said at least one storage means is thereby preferably controlled by respective synchronization signals, i. e. synch signals, that are contained within said input video signal, said modified video signal, said processed video signal, and/or said output video signal.

[0042] The inventive signal processing system advantageously comprises at least one synch conversion module adapted for detecting a synch signal in said input video signal and/or said processed video signal, and providing a field restart signal to said storage means wherein said storage means is adapted for receiving said field restart signal, and using said field restart signal for controlling reading and/or writing from/to said storage means.

[0043] A computer program product according to the invention comprises computer program means adapted to perform and/or to realize the method of video signal processing as defined above and/or the steps thereof when it is

8/25/06, EAST Version: 2.1.0.14

executed on a computer, a digital signal processing means, and/or the like.

[0044] An inventive computer readable storage medium comprises a computer program product as defined above.

[0045] The invention and advantageous details thereof will be explained by way of an exemplary embodiment thereof in the following with reference to the accompanying drawings, in which

Fig. 1 shows the first embodiment of the invention where an NTSC input video signal is processed by a PAL video processing core;

Fig. 2 shows a diagram illustrating the differences between the intermediate format and the first or input format;

Fig. 3 illustrates the reading and writing from/to the expansion buffer, and the compression buffer, respectively; and

Fig. 4 shows the second embodiment of the invention where a PAL input video signal is processed by an NTSC video processing core.

[0046] In Fig. 1 an input video signal 1 is written into an expansion buffer EB. In the embodiment of Fig. 1, the above mentioned first embodiment, the input video signal is of the NTSC format. This means, 858 pixels per line are written into the expansion buffer EB. As soon as pixels from the input video signal 1 are written into the expansion buffer EB, a PAL processing core PAL-PC starts reading pixels from the expansion buffer EB. After the first line of the input video signal 1, i. e. an NTSC video line, has been written into the expansion buffer EB and read-out with the same pixel rate by the PAL processing core PAL-PC, six blank data pixels i. e. added blank pixels ABP are generated and added to the first NTSC video line. During the processing of these 6 blank data pixels by the PAL processing core PAL-PC pixels of the next NTSC input line arrive, i. e. more pixels are provided by the input video signal 1. These pixels, i. e. six pixels, are stored in the expansion buffer EB. After the PAL processing core PAL-PC has processed the first entire NTSC video line plus the six blank data pixels which have been added, the PAL processing core PAL-PC starts processing the next NTSC input line. This procedure of writing to/reading from the expansion buffer EB continues until a video field of the input video signal 1 has been written to the expansion buffer EB. Due to the unbalanced read/write ratio at the input of the expansion buffer EB and the output of the expansion buffer EB, respectively, the expansion buffer EB fills up during the processing of one video field. In particular, the expansion buffer EB fills up by 6 pixels for each arrived video line of the input video signal 1.

[0047] It should be noted again, that the invention is based on the assumption of an equal number of processed pixels per time instance, i. e. an equal pixel rate for both formats, i. e. here the PAL format, which is the processing format, and the input format, which here is the NTSC format. In the embodiment of Fig. 1, the source format NTSC has 858 pixels per line and 263 lines per field, which equals 225654 pixels per field. The core processing format, i. e. the PAL format, has 864 pixels per line. Thus, only 225654 / 864 = 261.1 lines per NTSC field will be processed. This includes all required 244 lines of active video in NTSC. Note, that not all data needs to be written. Only active video data can be written to the buffer, thus reducing the maximum buffer size, i. e the maximum needed size of the buffer.

[0048] After one video field of the input video signal 1 has been written to the expansion buffer EB, an embedded synch signal, i. e. a field restart signal FR, is detected in the input video signal 1 by a first synch conversion module SC-1. The first synch conversion module SC-1 detects the field restart signal FR contained in the input video signal 1 and sends it to the PAL processing core PAL-PC and to the expansion buffer EB. As soon as the PAL processing core PAL-PC receives the field restart signal FR, the PAL processing core PAL-PC interrupts the processing of a current video field currently being processed and starts processing pixels from the following video field. The field restart signal FR also controls the writing component of the expansion buffer EB. The consequence is, that pixels remaining in the expansion buffer EB and belonging to the current video field are discarded. Discarding here can also mean that the remaining pixels in the expansion buffer EB are overwritten by new incoming data, i. e. pixels received via the input video signal 1.

[0049] The first synch conversion module SC-1 also translates the synch of the input video signal 1 into the processing synch pattern, i. e. the field restart signal of the processing format, here the PAL format, from which a field restart condition is derived, and controls the reading component of the expansion buffer EB. The field restart is synchronous to the INOUT (NTSC) format, thus the PAL case is interrupted in a premature manner.

[0050] At the output of the PAL processing core PAL-PC a processed video signal 3 is output. Each video line of the processed video signal 3 contains at the end of each line processed blank pixel PBP. In the embodiment of Fig. 1 these processed blank pixels PBP correspond to the six added blank pixels ABP. The processed blank pixels PBP are now removed from the processed video stream, i. e. the processed video signal 3. The remaining processed pixels are written to a compression buffer CB. At the output of the compression buffer CB an output video signal 4 is output. The output of the output video signal 4, however, does not start until the compression buffer CB contains a predetermined minimum number of pixels MNP. In the embodiment of Fig. 1 this minimum number of pixels MNP corresponds to the

8/25/06, EAST Version: 2.1.0.14

number of pixels contained in 1.8 video lines of the input video signal. During the time the compression buffer CB is filled with processed video pixels of the processed video signal 3 blank pixels will be output by the output video signal 4. The number of output blank pixels in the output video signal 4 is equal to the number of added blank pixels ABP per video field. In other words, all processed blank pixels PBP are output in a single block, i. e. sequentially. After the compression buffer CB is filled with the minimum number of pixels MNP, reading from the compression buffer CB starts in a FIFO-manner, i. e. pixels which have been stored in the buffer first are output first.

[0051] The second synch conversion module SC-2 does something similar as the first synch conversion module SC-1. The core processing module generates a new synch signal, which is embedded in the processed video signal 3. Here, the new synch signal controls the writing component of the compression buffer CB by a field restart write signal FRW. The second synch conversion module SC-2 also provides a field restart read signal FRR, which controls the reading component of the compression buffer CB. The output data stream of the compression buffer CB contains the new embedded synch signal, i. e. the field restart read signal FRR. This means, the field restart write signal FRW is synchronized with the processed video signal 3 of the intermediate format PROC, and the field restart read signal FRR is synchronized with the output video signal 4 of the NTSC format. Since it is generated by SC-2 it is guaranteed that the output data stream contains the field restart read signal FRR. Fact is that FRW causes FRR which are both caused by FR(SC-1). Thus all FR signals are synchronous to the IO format and not to the format PROC.

[0052] Fig. 2 shows a diagram illustrating the differences between an input format INOUT and an intermediate format PROC. It should be emphasized that the intermediate format PROC is not a valid standard video format, i. e. with respect to Fig. 1, the intermediate format PROC is not a valid PAL signal and not a valid NTSC signal.

[0053] As can be seen in Fig. 2 the input format INOUT, in the first embodiment corresponding to the NTSC format, has an input format width INOUT-FW and an input format height INOUT-FH. The input format width INOUT-FW and the input format height INOUT-FH correspond to one video field of the input video signal of the input format INOUT, here the NTSC format. As can be seen, one video field of the input format INOUT contains active video data that corresponds to an active video width AVW and an active video height AVH. Further, a video field of the input format INOUT contains horizontal and vertical blanking information HVB, which is all the video data of the video field that is not active video. Fig. 2 also shows an intermediate Hblank data width PROC Hblank of the intermediate format PROC and an input/output Hblank data width INOUT Hblank of the input format INOUT. The input/output Hblank data width INOUT Hblank of the INOUT format plus a patch data width is equal to the intermediate Hblank data width PROC Hblank.

[0054] Fig. 2 also shows a video field of the intermediate format PROC. This intermediate format PROC has an intermediate format width PROC-FW and an intermediate format height PROC-FH. As can be seen, the intermediate format width PROC-FW is larger than the input format width INOUT-FW, and the intermediate format height PROC-FH is smaller than the input format height INOUT-FH. Each line of the intermediate format PROC contains a complete line of the input format INOUT plus patch data PD. In the first embodiment one line of the intermediate format PROC contains a complete NTSC video line corresponding to 858 pixels plus six pixels of patch data PD.

[0055] The video field in Fig. 2 corresponding to the intermediate format PROC is completely processed by the PAL processing core PAL-PC. As can be seen in Fig. 2, thus all active video contained in the field of the input video signal having the input format INOUT will be processed. Thus, the processed video signal 3 contains processed active video PAV that corresponds to active video of the input video signal that has been processed by the PAL processing core PAL-PC.

[0056] Fig. 2 also shows unprocessed input data UID, which corresponds to data of the input video signal 1 that is contained in the expansion buffer EB after one video field of the input video signal 1 has been received, i. e. written into the expansion buffer EB. After the processing of a video field of the intermediate format PROC the field restart signal FR is extracted from the input video signal 1 by the first synch conversion module SC-1 and a new video field of the input video signal 1 will be processed by the PAL processing core PAL-PC. However, as can be seen in Fig. 2 the unprocessed input data UID does not contain any active video data of the video field of the input video signal 1. In other words, the unprocessed input data UID only contains horizontal and vertical blanking information HVB of the input video signal 1. It is important to note that all active video of the input video signal 1 has been properly processed during one cycle, i. e. during the processing of one video field of the input video signal 1.

[0057] Fig. 3 is still related to the first embodiment of the invention and shows in the upper part the writing and reading to/from the expansion buffer EB. The diagrams show progressed video lines PL over time. In the beginning, i. e. for t = 0, the expansion buffer EB is empty. At t = 0 the first field line of the input video signal 1, i. e. the first video line is an NTSC video line and will be processed by the PAL processing PAL-PC in real time, i. e. with the same pixel rate that pixels arrive with. Because of the same processing rate of the PAL processing core PAL-PC and the pixel arriving rate of the input video signal 1 during the processing of the first field line of the input video signal 1 no pixels need to be stored in the expansion buffer EB. After the processing of the first field line six added blank pixels ABP are inserted into the pixel stream at the end of the expansion buffer EB. During the insertion of these added blank pixels ABP new pixels arrive via the input video signal 1. Since the PAL processing core PAL-PC operates with the same pixel rate as the arrival pixel rate, during the processing of the second field line of the input video signal 1 six pixels will be stored

8/25/06, EAST Version: 2.1.0.14

in the expansion buffer EB. As can be seen in the upper part of Fig. 3, due to the unbalanced write/read ratio the expansion after EB will be filled and at the end of the processing of one video field of the input video signal 1 buffered pixels BP remain in the buffer, i. e. after a last field line LFL of the input video signal 1 has been written into the expansion buffer EB. After one video field of the input video signal 1 has been received, a field restart signal FR is received. Then, a new video field of the input video signal 1 will start to be processed. Therefore, the pixels that still remain in the buffer will be deleted. The number of pixels that remain in the buffer after the receiving of one video field is equal to the minimum number of pixels MNP. The minimum number of pixels MNP corresponds to the number of added blank pixels ABP per video field. The minimum number of pixels MNP can be calculated as follows: An NTSC field time unit, i. e. the number of pixels per NTSC video field is 858 x 263 clocks, i. e. 263 source lines; this field time unit is equal to 858 x 263/864 = 261.1 processed video lines; this means, 263 source lines, i.e. video lines contained in the input video signal 1, are stored, but only 261.1 video lines, i. e. PROC lines i. e. lines of the intermediate format are read by the PAL processing core PAL-PC. Therefore, roughly 2 lines remain in the buffer, which correspond to the minimum number of pixels MNP. The number of buffer lines, i. e. the number of video lines that remain in the expansion buffer EB at the end of one video field can be calculated by the following equation:

$$\frac{\sum line_patch_data}{\frac{number_of_input_lines}{PROC_line_length}} = number_of_buffer_lines \qquad (1)$$

the variables denoting:

line_patch-data:	the number of added blank pixels ABP per video line,
number_of_Input_lines:	the number of input lines of the video input signal 1 per video field.
PROC_Line_length:	the line length , i. e. the number of pixels per line of the intermediate format PROC,
number_of_buffer_lines:	the number of lines remaining in the buffer after one video field has been received, i. e. after the restart signal FR has been received.

[0058] Equation (1) helps computing the buffer size when (a) storing an entire video line including a blank data or when (b) only storing active video.

[0059] If only active video is stored with 720 pixels per line, the buffer size can be reduced. The maximum needed size of the buffer, i. e. if 864 pixels are processed is given by:

1.82 lines x 864 pix and 16 bit each = 24.6 kbit per buffer (maximum).

and the minimum needed buffer size, i. e. when only active video is stored is given as follows:

1.82 lines x 720 pix and 16 bit each = 20.5 kbit per buffer (minimum).

[0060] Since per line only 720 pixels need to be stored by skipping Hblank data, only the number of lines to be stored (e. g. 1,82) multiplied by 720 instead of 864 pixels with 16 bit per pixel equals the needed buffer size. This reduces the buffer size requirements significantly.

[0061] The lower part of Fig. 3 shows the writing W and reading R to/from the compression buffer CB. At t = 0 a first field line of processed video signal FFLP is written to the compression buffer CB. This is controlled by the field restart write signal FRW. Subsequently, further field lines of processed video signal are written to the compression buffer until the minimum number of pixels MNP has been written to the compression buffer CB corresponding to a buffer fill time BFT. At the buffer fill time BFT, reading R starts from the compression buffer CB. The reading is controlled by the field restart read signal FRR. Reading R starts at a higher rate such that the number of buffered pixels BP continually drops until the compression buffer CB is empty at the end of one video field, i. e. when a field restart write signal FRW is provided by the second synch conversion module SC-2. At this time the last field line of processed video signal LFLP has been read from the compression buffer CB. Note, that in the lower part of Fig. 3 writing and reading does not start synchronously as is the case in Fig. 3 upper part. In Fig. 3 lower part, when the buffer is empty, only blank pixels are put out until the next read restart pulse, i. e. the field restart read signal FRR, re-triggers the reading process. The

8/25/06, EAST Version: 2.1.0.14

blank pixels that are output, i. e. the blank lines, can be considered as all horizontal patch data being put out in a single block of about 1.8 lines of video.

[0062] Fig. 4 shows the second embodiment of the invention where an input video signal 1 of the PAL standard is received and the processing core is designed for processing NTSC format video signals, i. e. an NTSC processing core NTSC-PC. In this embodiment a compression buffer CB is used at the input of the NTSC processing core NTSC-PC. Before pixels of the video input signal 1 are stored in the compression buffer CB excess pixels EP are removed from the input video signal 1. This is because the input video signal 1 in this embodiment is of the PAL standard having 864 pixels per line whereas in this embodiment the intermediate format PROC only has 858 pixels per line which is equal to the number of pixels per line of the processing format, i. e. here the NTSC format. The excess pixels EP correspond to pixels of the Hblank and/or the Vblank period of the input video signal 1. Since in this embodiment reading of the NTSC processing core NTSC-PC from the compression buffer CB starts at a higher rate than writing to the compression buffer CB, no pixels are read from the compression buffer CB until a second minimum number of pixels 2MNP are stored in the buffer, i. e. the buffer is filled with the minimum number of pixels 2MNP.

[0063] Note, that the buffer size for the second embodiment is different from the needed buffer size in the first embodiment. The buffer size of the second embodiment, which is equal to the second minimum number of pixels 2MNP may also be calculated using formula (1) from above. In the case of the second embodiment, the needed buffer size for the compression buffer CB is equal to 2.18 video lines.

[0064] In Fig. 4, the first synch conversion module SC-1 detects a field restart signal contained in the input video signal 1 and provides this field restart signal as a field restart write signal FRW to the compression buffer CB. This means, the field restart write signal FRW is synchronous with the field restart signal of the input video signal 1 of the PAL format. The first synch conversion module SC-1 also provides a field restart read signal FRR, which is also synchronous to the field restart signal of the input/output or INOUT video signal 1 of the format INOUT. In other words, the first synch conversion module SC-1 provides the field restart write signal FRW and the field restart read signal FRR to the compression buffer CB in order to control the writing to reading from the compression buffer CB. As can be seen in Fig. 4, the field restart read signal FRR is also provided to the NTSC processing core NTSC-PC. Here, the field restart read signal FRR is used to control the field-wise processing of the NTSC processing core NTSC-PC.

[0065] It should be noted that the field restart write signal FRW and the field restart read signal FRR are used in an analogue way to the first embodiment of above. This means, the explanations of Fig. 3 lower part also apply here. As can be seen from Fig. 3 lower part, writing to the compression buffer CB starts when the first synch conversion module SC-1 provides the field restart write signal FRW at t = 0, and reading does not start until the first synch conversion module SC-1 provides the field restart read signal FRR. This is the case when the compression buffer CB is filled with the second minimum number of pixels 2MNP. At the time the next field restart write signal FRW is provided by the first synch conversion module SC-1, the compression buffer CB will be empty. This is due to the higher rate for reading R pixels from the compression buffer CB than the rate for writing W to the compression buffer CB. As in the first embodiment, it is always assured that all active video will be processed by the NTSC processing core NTSC-PC.

[0066] At the output of the NTSC processing core NTSC-PC processed video pixels of the intermediate format PROC are written to the expansion buffer EB. As soon as pixels have been written to the expansion buffer EB pixels are output to yield the output video signal 4. However, reading starts with a lower rate, because at the end of each video line of the processed video signal 3 which is read from the expansion buffer EB blank pixels will be added, i. e. added blank pixels ABP. This way the output video signal 4 will have 864 pixels per line as is required for the PAL standard. At the end of a such generated video line, i. e. the pixels of a line of the processed video signal 3 plus the 6 added blank pixels ABP, the second synch conversion module SC-2 provides an according synch signal, i. e. a field restart signal FR. Thus, the output video signal 4 is a video signal completely conform with the PAL standard.

[0067] In the following further elucidations are provided that may help a person skilled in the art to get a better understanding of the invention.

[0068] A video image processing core module works for a specific image format which is mainly determined by the format width. By using expansion buffers (slow-down buffers) and compression buffers (speed-up buffers) at the respective inputs and outputs of the system, a format width conversion is achieved. To expand the width, data is patched into the data stream, whereas data is skipped from the stream to compress the image width. Thus, the interface format, i. e. the intermediate format, is matched to the processing core format. The invention does provide a simple method that uses relatively little memory and logic and is fully transparent to the core module's format.

[0069] The invention may be referred to as multi-format adaptation scheme and can be used in an ASIC as part of a commercial TV product. As already mentioned, with a relatively small amount of additional memory and logic added to the system, e. g. a respective ASIC, other formats, i. e. formats that are different from the formats the processing core, 1. e. the system, is designed for, are added to the acceptable input/output format list without changing the core processing module. This relaxes the designer's requirements to evaluate and verify the core module for other formats. Only the format adaptation needs to be verified.

[0070] One advantage of the invention is that once a PAL processing core is designed, tested and verified, by only

8/25/06, EAST Version: 2.1.0.14

adding some small buffers at the interfaces, the NTSC processing capability is also given. Without the invention a processing algorithm which is comparable with both formats would be needed. However, keeping an implementation of such an algorithm generic in terms of format width, is very difficult and time consuming. Also, legacy implementations might not be available for other than one format. These problems do not arise when the invention is used, providing multiple format compatibility with low memory requirements.

[0071] According to the invention, in order to process an input image, e. g. 858 pixels wide which correspond to NTSC with a processing module designed for a different width, e. g. 864 pixels wide which correspond to PAL standard, the incoming image is stored into an expansion (slow-down) buffer. The width adaptation is done by reading the stored image data from the expansion (speed-up) buffer and adding additional patch data to the image to do the width conversion. The output of the processing module is then stored into a compression buffer where the reverse width adaptation is done by skipping the previously inserted patch data. This example describes the narrow input/output format with a wider core processing format, i. e. the first embodiment of the invention.

[0072] In the second embodiment of the invention the wider input/output format is first compressed (speed-up), processed in the narrow format and then expanded (slow-down).

[0073] In the first embodiment the maximum memory requirement of the compression buffer CB and the expansion buffer EB is 49.2 kbit buffer memory and is computed as follows:

- With 261.1 lines of PAL processed data per 263 lines NTSC, roughly 1.82 lines of active video need to be stored at maximum in the expansion (slow-down) buffer. This is 1.82 x 864 pixels each 16 bit = 24.6 kbit. That is, after one NTSC input field, the expansion buffer is filled.
- The same amount of data needs to be stored into the compression (speed-up) buffer before read-out starts. That is, after one NTSC output field, the compression buffer is empty.

[0074] In the following the first embodiment will be explained in more detail. In this case an NTSC format is used as input video signal having 858 pixels per line, 262 or 263 lines, and 60 fields per second. The core module is designed for the PAL format having 864 pixels per line, 313 or 312 lines, and 50 fields per second.

[0075] The following points should be noted:

- A processing core is designed to work on a specific line width, determined by the core format. The invention changes the INOUT format line length to match the processing format line length.
- The number of lines per video field to be processed by the core is determined by a respective field restart signal (video synch signal) and not by core format line number. E.g. a PAL processing core can be interupted and restarted before reaching 312 lines, whereas a NTSC processing core can process more than 262 lines.
- The pixel rate of the INOUT format is identical to the pixel rate of the processing format.

[0076] The expansion and compression buffers are located around the core processing module as depicted in Fig. 1. This block diagram is a structural representation. The actual implementation merges the respective buffers with video memories inside the processing core. But processing core memory is not needed for the invention to work.

[0077] The PAL core processing format has 864 pixels per line and the NTSC I/O format has 858 pixels per line, which is 6 pixels difference. The NTSC data will be written into the expansion buffer. Reading the buffer starts immediately after writing starts, in order to keep the buffer size small and the latency of the system small. After writing/reading the first 858 pixels (one source video line), 6 blank pixels are patched into the intermediate format stream to complement a video line. Meanwhile, more source data is written into the expansion buffer. Due to the unbalanced write/read ratio, the buffer will fill up during one full field cycle.

[0078] The previous description is based on a full video line, which contains active video and also horizontal blanking data (Hblank). Writing the entire NTSC video line with 858 pixels is only needed, when the Hblank data needs to be transmitted through the system. If only the active video data is needed, merely 720 pixels per video line need to be stored. This Hblank data requirements thus affect the needed buffer depth, which computes as follows:

- An NTSC field time unit is 858 x 263 clocks (=263 source lines).
- This time unit is equal to 864x261.1 clocks (858 x 263/864 = 261.1 proc lines).
- That means, 263 source lines are stored, but only 261.1 proc lines are read → roughly 2 lines remain in the buffer.
- This is the sum of all patch data divided by the proc format line length:

8/25/06, EAST Version: 2.1.0.14

$$\frac{\sum line_patch_data}{\frac{number_of_input_lines}{PROC_line_length}} = number_of_buffer_lines$$

- Thus, the buffer stores roughly 2 (= 1.82) video lines. If only active video is stored with 720 pixels per line, rather than the entire 864 pixels, the buffer size can be reduced.
- 1.82 lines x 864 pix and 16 bit each = 24.6 kbit per buffer (maximum).
- 1.82 lines x 720 pix and 16 bit each = 20.5 kbit per buffer (minimum).

[0079] As mentioned before, both formats, i.e. the input format INOUT and the intermediate format PROC have the same pixel rate. The "temporal field rate" and the "number of lines per field" are parameters in a respective format. With an equal field rate for both formats, wider fields mean fewer lines per field, or vice versa. The width of a field represents the active video component plus the horizontal blanking period. Also, with an equal number of lines per field for both formats, wider lines mean a lower field rate, respectively.

[0080] See Fig. 2 for a visualization. If a wider intermediate format assume, less lines of the INOUT format will be processed, given an equal field rate. In our preferred embodiment the overlapping area covers the entire active video of the INOUT format.

[0081] In our case, though NTSC has also fewer lines than PAL, but the field rate is higher. From the above calculation for the buffer depth, it can be seen that during an NTSC field, 260 lines of PAL converted video is processed. This is more than the minimum of 244 lines of active video in NTSC. Thus, the buffer depth can be reduced even, such that processing stops when 244 lines are processed.

[0082] On the output side of the processing core, the PAL signal has to be reformatted to match the INOUT format. Writing into the compression buffer starts, as soon as the processing module provides valid data. Then, when the compression buffer is filled, reading starts with a higher rate. For every 864 pixels being processed, 858 pixels are written and 6 pixels are discarded. When the entire buffer is empty the NTSC field has been written out. This concludes one full cycle.

[0083] Fig. 3 shows the utilization of the respective buffers. After one cycle the expansion buffer at the system input is filled and will be deleted when the next cycle starts. That is, new incoming data overwrites old data. On the other end the compression buffer at the system output needs to be filled, before reading of data can start. During this time, blank data will be put out to the INOUT format. This blank data can be considered as the entire patch data being sent in a single block. In the first embodiment, this is about 2 lines of video data.

[0084] Finally, it should be noted that the invention helps to save a lot of development time and reduces design risks by reusing an existing and already tested processing core design. The video signal processing core, i. e. the processing core design, used is not restricted to special functionality, but it can do video processing of any type.

Reference Symbols

[0085]

1	Input video signal
2	modified video signal
3	processed video signal
4	output video signal
PROC	intermediate format
INOUT	first or input format
EP	excess pixels
FR	field restart signal
FRW	field restart write signal
FRR	field restart read signal
HVB	horizontal and vertical blanking information
PAL-PC	PAL processing core
NTSC-PC	NTSC processing core
PD	patch data
EB	expansion buffer
CB	compression buffer
ABP	added blank pixels

8/25/06, EAST Version: 2.1.0.14

PBP	processed blank pixels
SC	synch conversion module
SC-1	first synch conversion in module
SC-2	second synch conversion in module
INOUT-FW	input format width
INOUT-FH	input format height
PROC-FW	intermediate format width
PROC-FH	intermediate format height
PROC Hblank	intermediate Hblank data width
INOUT Hblank	input/output Hblank data width
BP	buffered pixels
LFL	last field line of input video signal
FFL	first field line of input video signal
LFLP	last field line of processed video signal
FFLP	first field line of processed video signal
BFT	buffer fill time
MNP	minimum number of pixels
PL	progressed lines

Claims

1. Method for video signal processing, wherein the following steps are performed:

   - receiving an input video signal (1) having a first or input format (INOUT, NTSC; INOUT, PAL),
   - processing said input video signal (1) or a modified video signal (2), which is a derivative of said input video signal (1), using a video signal processing method, which is designed for processing video signals of a fixed second or processing format (PAL; NTSC) thus generating a processed video signal (3),
   - outputting said processed video signal (3) or a derivative thereof as an output video signal (4),
   - wherein said modified video signal (2) is derived from said input video signal (1) by adding or removing pixels to/from said input video signal (1), thereby obtaining said modified video signal (2) with an intermediate format (PROC), if said first or input format (INOUT, NTSC; INOUT, PAL) is not conform to said fixed or processing second format,
   - wherein said output video signal (4) is derived from said processed video signal (3) by accordingly removing or adding pixels from/to said processed video signal (3) thereby obtaining an output video signal (4) of said first format, if said first or input format (INOUT, NTSC; INOUT, PAL) is not conform to said second or processing format (PAL; NTSC).

2. Method according to claim 1, **characterized in that**, said intermediate format (PROC) is not a valid standard video format, in particular it is neither of the first or input format (INOUT, NTSC; INOUT, PAL) nor of the fixed second or processing format (PAL; NTSC).

3. Method according to claim 1 or 2, **characterized in that** the number of lines of said intermediate format (PROC) is equal to the number of pixels per video field of said first or input format (INOUT, NTSC; INOUT, PAL) divided by the number of pixels per video line of said second or processing format (PAL; NTSC).

4. Method according to any one of the preceding claims, **characterized in that**, said first or input format (INOUT, NTSC; INOUT, PAL) is a first standard video format (NTSC; PAL) and/or said fixed second or processing format (PAL; NTSC) is a second standard video format (PAL; NTSC).

5. Method according to any one of the preceding claims, **characterized in that**, when deriving said modified video signal (2) from said input video signal (1) pixels are added/removed so as to change the number of pixels per video line (PROC-FW) of said intermediate format (PROC) with respect to said first or input format (INOUT, NTSC; INOUT, PAL).

6. Method according to claim 4 or 5, **characterized in that**, pixels are added/removed to/from said input video signal (1) such that the number of pixels per video line (PROC-FW) of said intermediate format (PROC) is equal to the number of pixels per video line of said fixed second or processing format (PAL; NTSC).

8/25/06, EAST Version: 2.1.0.14

7. Method according to any one of the preceding claims, **characterized in that**, when deriving said output video signal (4) from said processed video signal (3) said pixels are removed/added so as to change the number of pixels per video line (INOUT-FW) of said output video signal (4) with respect to said processed video signal (3).

8. Method according to claim 7, **characterized in that**, pixels are removed/added from/to said processed video signal (3) such that the number of pixels per video line (INOUT-FW) of said output video signal (4) is equal to the number of pixels per video line of said first or input format (INOUT, NTSC; INOUT, PAL).

9. Method according to any one of the preceding claims, **characterized in that**, said video signal processing method uses the field restart signal (FR, FRW, FRR) of said input video signal (1), such that the field processing rate of said processing method is equal to the field rate of the first or input format (INOUT, NTSC; INOUT, PAL).

10. Method according to any one of the preceding claims, **characterized in that**, said video processing method is performed by a video processing core (PAL-PC; NTSC-PC), which is in particular adapted/designed for said fixed second or processing format (PAL; NTSC).

11. Method according to any one of the preceding claims, **characterized in that**, said first or input format is the NTSC video standard and said second or processing format (PAL; NTSC) is the PAL video standard.

12. Method according to claim 11, **characterized in that**, blank pixels (PD) are added at the end of an input video signal (1) line thus obtaining said modified video signal (2) having said intermediate format (PROC).

13. Method according to claim 11 or 12, **characterized in that**, pixels of said input video signal (1) are stored in a first buffer (EB) and said processing method reads pixels from said first buffer (EB).

14. Method according to any one of claims 11 to 13, **characterized in that**, processed pixels of said processed video signal (3) which correspond to added pixels (PD) of said modified video signal (2) are removed from said processed video signal (3) thus obtaining said output video signal (4).

15. Method according to any one of claims 11 to 14, **characterized in that**, pixels of said processed video signal (3) are stored in a second buffer (CB) and said output video signal (4) is generated by reading pixels from said second buffer (CB).

16. Method according to any one of claims 11 to 15, **characterized in that**, a first buffer (EB) and/or a second buffer (CB) are used having a capacity to at least store a minimum number of pixels (MNP), which is determined by the number of added pixels (PD) per video field of said input video signal (1).

17. Method according to claim 16, **characterized in that**, reading of pixels from said second buffer (CB) does not start until said second buffer (CB) is filled with at least said minimum number of pixels (MNP).

18. Method according to claim 16, **characterized in that**, reading of pixels from said second buffer (CB) does not start until a field restart signal (FRR) is provided.

19. Method according to any one of claims 1 to 10, **characterized in that**, said first or input format (INOUT, NTSC; INOUT, PAL) is the PAL video standard and said second or processing format (PAL; NTSC) is the NTSC video standard.

20. Method according to claim 19, **characterized in that**, pixels are removed at the end of an input video signal (1) line thus obtaining said modified video signal (2) having said intermediate format (PROC).

21. Method according to claim 20, **characterized in that**, removed pixels correspond to pixels of the horizontal blanking period.

22. Method according to any one of claims 19 to 21, **characterized in that**, pixels of said input video signal (1) are stored in a second buffer (CB) and said processing method reads pixels from said second buffer (CB).

23. Method according to any one of claims 19 to 22, **characterized in that**, blank pixels are added at the end of each video line of said processed video signal (3) thus obtaining said output video signal (4) of said first or input format

8/25/06, EAST Version: 2.1.0.14

(INOUT, PAL).

24. Method according to any one of claims 19 to 23, **characterized in that**, pixels of said processed video signal (3) are stored in a first buffer (EB) and said output video signal (4) is generated by reading pixels from said first buffer (EB).

25. Method according to any one of claims 19 to 24, **characterized in that**, a second buffer (CB) and/or a first buffer (EB) are used having a capacity to at least store a minimum number of pixels (MNP), which is equal to the number of pixels removed from said input video signal (1) per video field.

26. Method according to claim 25, **characterized in that**, said processing method does not start reading pixels from said second buffer (CB) until said second buffer (CB) is filled with at least said minimum number of pixels (MNP).

27. Method according to claim 25, **characterized in that**, said processing method does not start reading pixels from said second buffer (CB) until a field restart read signal (FRR) is provided.

28. Signal processing system, which is capable of performing or realizing a method for signal processing according to any one of the preceding claims 1 to 27 and/or the steps thereof.

29. Signal processing system according to claim 28, comprising

- a receiving means adapted for receiving an input video signal (1),
- an adding/subtracting means (+; -) adapted for adding/subtracting pixels to/from said input video signal (1) thus generating a modified video signal (2) having an intermediate format (PROC),
- a processing means (PAL-PC; NTSC-PC) designed for processing video signals of a fixed second or processing format (PAL; NTSC), which processing means (PAL-PC; NTSC-PC) is adapted for processing said modified video signal (2), and thus generating a processed video signal (3),
- a subtracting/adding means (-; +) adapted for subtracting/adding pixels from/to said processed video signal (3) thus generating an output video signal (4) of said first format, and
- an outputting means adapted for outputting said output video signal (4).

30. Signal processing system according to claim 28 or 29, **characterized by** at least one storage means (EB, CB; CB, EB) adapted for storing at least parts of said input video signal (1) and/or at least parts of said processed video signal (3).

31. Signal processing system according to any one of claims 28 to 30, **characterized by** at least one synch conversion module (SC-1, SC-2) adapted for detecting a synch signal in said input video signal (1) and/or said processed video signal (3), and providing a field restart signal (FR, FRR, FRW) to said storage means (EB; CB), wherein said storage means (EB, CB) is adapted for receiving said field restart signal (FR, FRR, FRW), and using said field restart signal (FR, FRR, FRW) for controlling reading and/or writing from/to said storage means (EB, CB).

32. Computer program product, comprising computer program means adapted to perform and/or to realize the method of video signal processing according to any one of the claims 1 to 27 and/or the steps thereof, when it is executed on a computer, a digital signal processing means, and/or the like.

33. Computer readable storage medium, comprising a computer program product according to claim 32.

8/25/06, EAST Version: 2.1.0.14

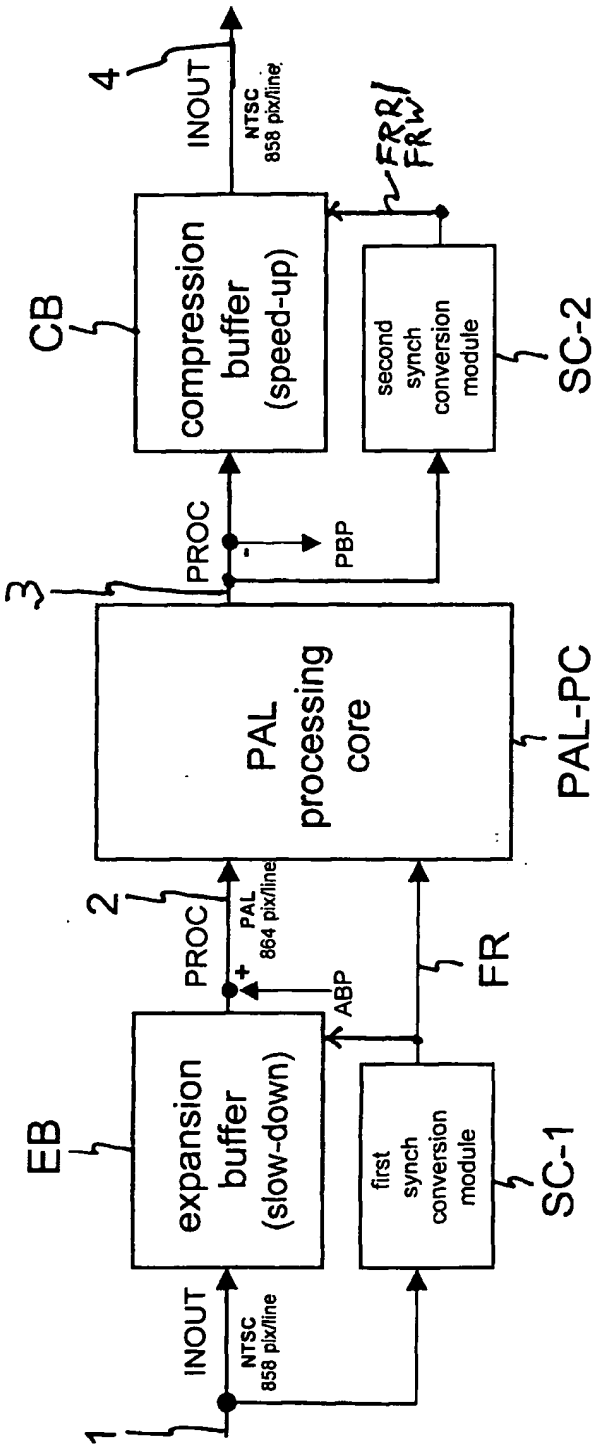


Fig. 1

8/25/06, EAST Version: 2.1.0.14

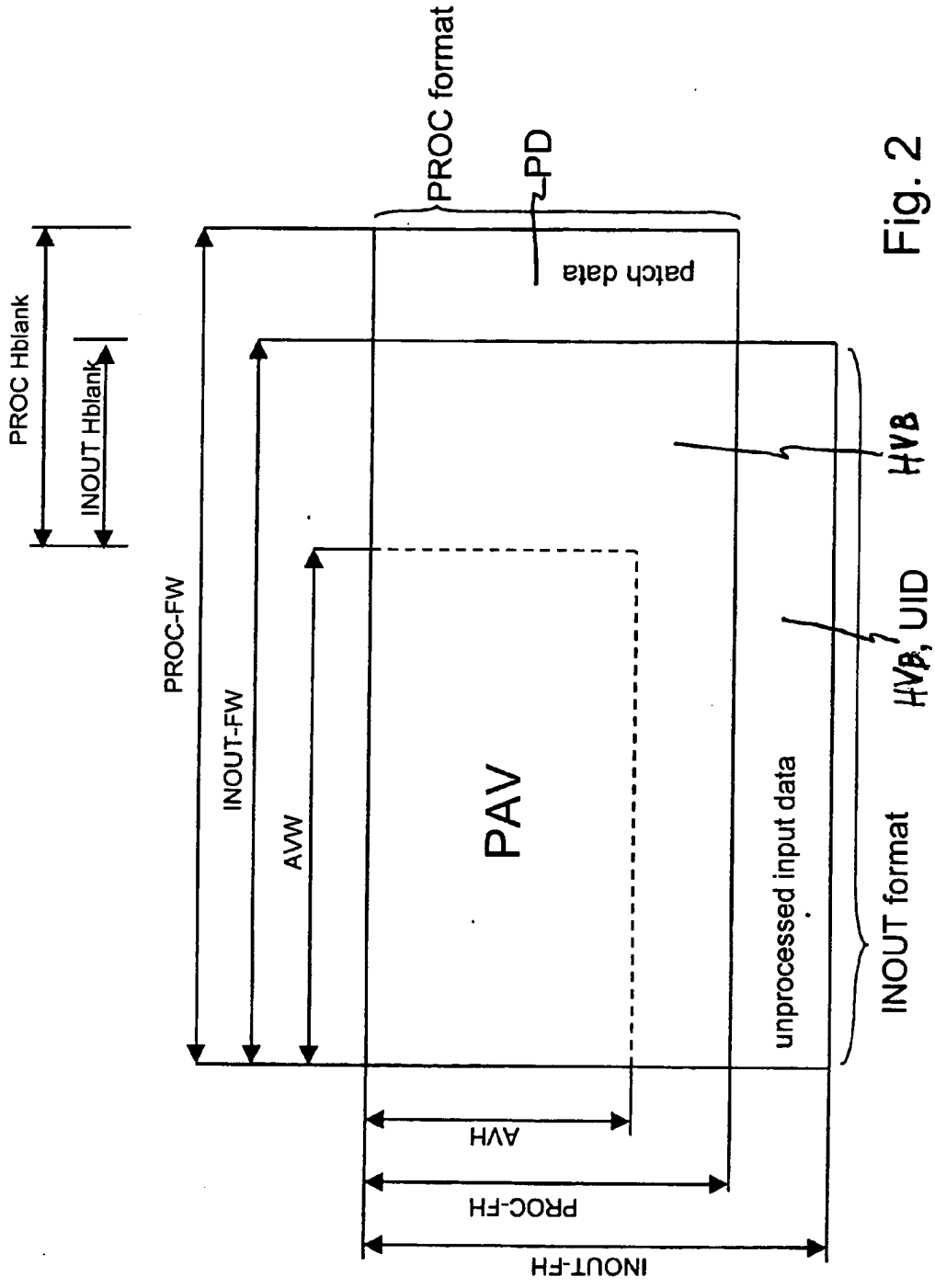


Fig. 2

8/25/06, EAST Version: 2.1.0.14

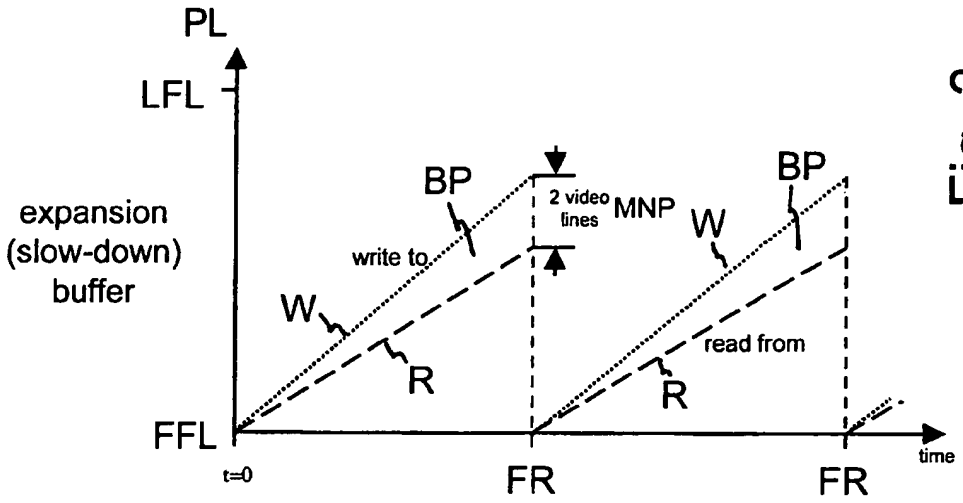
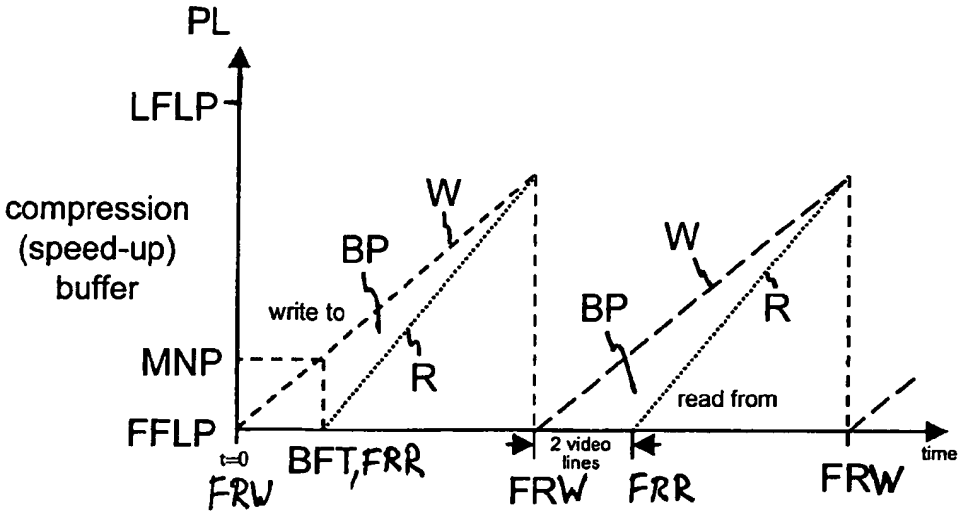

PL
LFL
expansion
(slow-down)
buffer
FFL
t=0
BP
2 video
lines
MNP
write to
W
R
FR
BP
W
read from
R
FR
time

Fig. 3


PL
LFLP
compression
(speed-up)
buffer
MNP
FFLP
t=0
FRW
BFT,FRR
W
BP
write to
R
2 video
lines
FRW
FRR
W
BP
R
read from
FRW
time

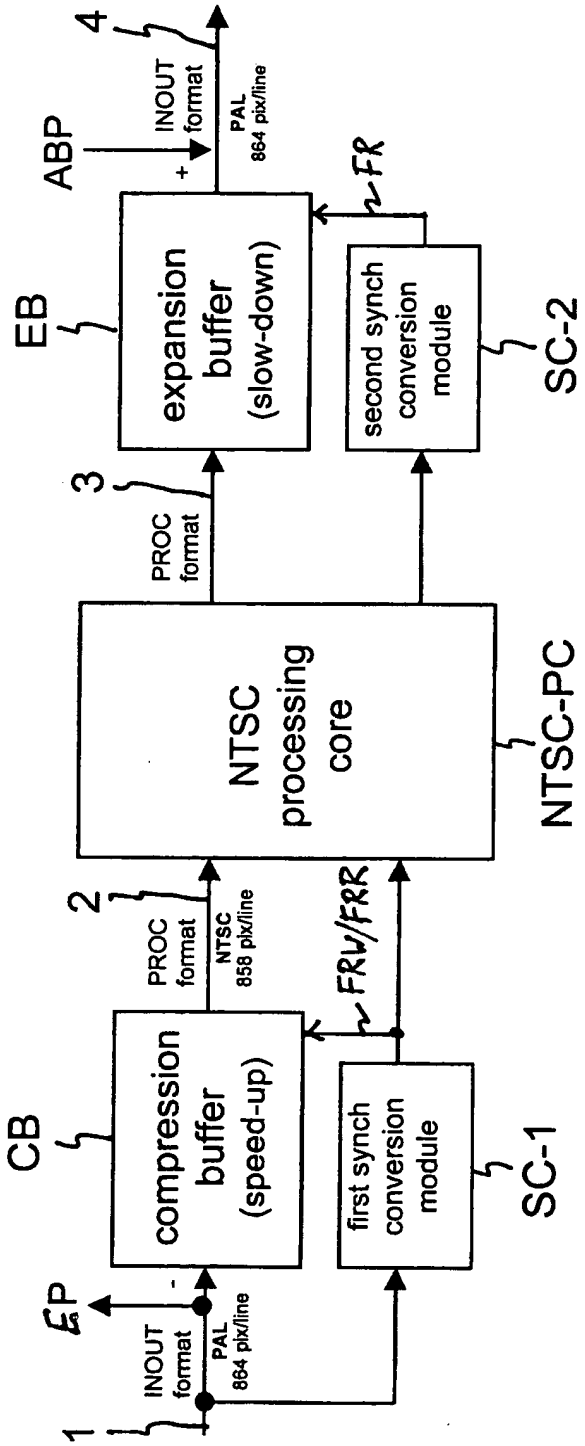

1
INOUT format
PAL 864 pix/line
EP
-
CB
compression buffer (speed-up)
2
PROC format
NTSC 858 pix/line
FRW/FRR
first synch conversion module
SC-1
NTSC processing core
NTSC-PC
3
PROC format
EB
expansion buffer (slow-down)
ABP
+
4
INOUT format
PAL 864 pix/line
FR
second synch conversion module
SC-2

Fig. 4

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 04 00 0556

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	US 5 990 982 A (GOVE ROBERT ET AL) 23 November 1999 (1999-11-23) * column 9, line 19 - line 28; figure 8 * -----	1,2,4-8, 28-30, 32,33	H04N5/46
A	EP 1 370 090 A (SONY INT EUROP GMBH) 10 December 2003 (2003-12-10) * the whole document * -----	1,28,32, 33	

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

H04N

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	22 June 2004	Yvonnet, J

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

8/25/06, EAST Version: 2.1.0.14

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 04 00 0556

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

22-06-2004

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5990982 A	23-11-1999	NONE	
EP 1370090 A	10-12-2003	EP 1370090 A1	10-12-2003

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

8/25/06, EAST Version: 2.1.0.14